REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-01-

0461

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED
02-08-01	Final Technical Report	15-04-98 - 30-11-00

4. TITLE AND SUBTITLE
Agents for Dynamic Plan Management

5a. CONTRACT NUMBER
F496290-98-1-0436

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Dr. Martha E. Pollack

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Dept. of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR
Dept. of the Air Force
801 North Randolph St., Rm. 732
Arlington, VA 22203-1977

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR/PKC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release,
distribution unlimited

20010905 127

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This project was aimed at developing agent-based technology for *dynamic plan management*, i.e., for automatically monitoring, extending, and updating complex plans for a human user in a changing environment in which the set of goals to be achieved evolves over time. The project succeeded in achieving its objectives, and obtained both theoretical and practical results. In particular, representations and algorithms were developed for four key plan management tasks: (i) *deliberation*; (ii) *plan merging*; (iii) *contingency selection*, and (iii) selective *execution monitoring*. Additionally, we designed and implemented a prototype plan management agent (PMA), which builds on these and other plan management algorithms, and assists a user in managing a complex set of plans.

15. SUBJECT TERMS
Automated Planning, Personal Assistants, Plan Management, Computational Agents

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Martha Pollack
unclassified	unclassified	unclassified		12	19b. TELEPHONE NUMBER (include area code) 734/615-8048

Air Force Office of Scientific Research
Final Technical Report
Contract Number F496290-98-1-0436

"Agents for Dynamic Plan Management"

Dr. Martha E. Pollack, Principal Investigator
Address during Contract:
Dept. of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260
Current Address:
Dept. of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, MI 48109
Period of Performance: 15 Apr 1998-30 Nov. 2000

Project Monitor: Dr. Alex Kilpatrick

Agents for Dynamic Plan Management

AFOSR Contract Number F496290-98-1-0436

Abstract

This project was aimed at developing agent-based technology for *dynamic plan management*, i.e., for automatically monitoring, extending, and updating complex plans for a human user in a changing environment in which the set of goals to be achieved evolves over time. The project succeeded in achieving its objectives, and obtained both theoretical and practical results. In particular, representations and algorithms were developed for four key plan management tasks: (i) *deliberation*, i.e., determining whether or not to adopt a new potential goal by assessing its cost in context; (ii) *plan merging*, i.e., integrating plans for new goals into a set of existing commitments, while minimizing the disruption to those prior commitments; (iii) *contingency selection,* i.e., making effective decisions about which contingent events to plan for in advance, and (iii) selective *execution monitoring*, i.e., determining which environmental features to attend to while a plan is being executed. Additionally, we designed and implemented a prototype plan management agent (PMA), which builds on these and other plan management algorithms, and assists a user in managing a complex set of plans.

Research Objectives

This research project was aimed at an important but challenging goal: that of developing foundations for intelligent agents that can help humans perform a range of information-processing and decision-making tasks more efficiently and effectively. In most cases, these agents will be deployed in dynamic, uncertain environments, where they will have to exhibit goal-directed behavior. Technology for generating plans to achieve goals in such environments has emerged, but the ability simply to generate plans in not sufficient. Agents in dynamic, uncertain environments must also be capable of effectively *managing* their plans. This research project was aimed at developing computational mechanisms—trepresentations, algorithms, and heuristics—for plan management tasks, and at demonstrating their viability in a prototype plan-management agent (PMA). The specific plan-management tasks that we modeled included:

- determining the cost of a new option in the context of an existing set of plans;
- determining whether a new option is compatible with a set of existing plans, and whether compatibility depends upon the imposition of additional constraints on the option's performance (e.g., about how or when the new option will be achieved), to enable plan update via merging;
- determining which possible plan failures to consider at plan-generation time, and which to defer to execution time; and
- determining what to monitor during plan execution, and at what times.

In addition to developing representations and algorithms for these tasks, we had a goal of implementing and integrating them in a prototype Plan Management Agent (PMA), a

system that has structured activities of a human user, and provides assistance in managing those activities. For example, PMA's capabilities include alerting a user to potential conflicts among his plans; suggesting plan expansions that have acceptable cost; and suggesting situations that should be monitored during plan execution. PMA would be valuable in a wide range of domain, including many military settings. For instance, a military commander overseeing a complex mission could be assisted by PMA, which would track personnel and equipment commitments, guarantee consistency among various operational goals, and guide reactions to new developments. It would need to be able to fit new plans into the ones to which it has already committed, and to determine whether its commitments remain viable after detected changes in the environment. It might also need to make decisions about planning detail, for instance, deciding whether a transportation plan should be completed now, or whether it is better to wait until the weather status of a certain airport is known.

Results

Here we briefly describe the main results of our research project. Our discussion is organized in terms of the four algorithmic issues mentioned in the previous section (i.e, the four main plan management tasks for which we developed representations and algorithms), and the overall system we designed and built. For each category, we list the relevant publications supported by this project.

Computing the Cost of a Plan in Context

References [2], [17]

A central reasoning task for any agent is to compute the expected value of an option for action that it is considering, to determine whether it is worthwhile committing to that option. The extensive literature on decision theory provides a mathematically rich basis for addressing this question, but falls short of our needs in designing an artificial agent in two ways. First, it assumes that the utility of an outcome is given as part of the background context of the decision problem. In fact, the overall desirability of an option presented to an agent is often not immediately apparent; and we were explicitly concerned with the mechanism through which it might be discovered. We have so far focused, in particular, on the case in which the option presented to an agent has a known benefit, but requires some effort—the execution of a plan—for its achievement. In order to evaluate the overall desirability of the option, the agent thus has to arrive at some assessment of the cost involved in achieving it.

Second, standard decision theory has not been concerned with computational constraints. In contrast, we have insisted that the task of evaluating an option should be computationally realizable; and in particular, our work has been developed within the theoretical framework proposed in our earlier research, according to which it is best to view a resource-bounded agent as operating always against the background of some current set of intentions, or plans. In contrast to standard decision theory, where actions are evaluated in isolation, we developed a model in which the options presented to an

agent are evaluated against a background context provided by the agent's current plans—commitments to future activities, which, at any given point, may themselves be only partially specified. The interactions between the new option and the background context can complicate the task of evaluating the option, rendering it either more or less desirable in context than it would have been in isolation.

As a commonsense example, suppose an agent is already committed to going to the airport tomorrow afternoon to catch a plane, but has not yet decided whether to get there by taxi or by taking the airport shuttle van. Given this background context, the agent might then have to evaluate the newly presented option of attending a lunchtime meeting tomorrow. If the meeting is to be held at the agent's office, and is likely to run late, a decision to attend may rule out the possibility of taking the van. Assuming that the van costs less than the taxi, the new option would then be less desirable in context that it would have been in isolation; the benefit of attending the meeting must be at least great enough to compensate for the difference in cost between taxi and van to make it worthwhile. On the other hand, suppose the meeting is to be held at an airport hotel. In this case, the background context reduces the cost associated with the new option, increasing its overall desirability, since the agent is already committed to going to the airport: the agent might rationally choose to attend the meeting, since he is going to the airport anyway, even if this option is not one the agent would have decided to pursue in isolation.

Our work on this problem provides an initial theoretical and computational analysis of the reasoning involved in situations like this, where a new option must be evaluated within the context provided by a background plan. Because we are concerned with the design of artificial agents, we represented both the agent's background context and the new options it might encounter using the well-understood plan formalisms common in AI. Our approach to evaluating the desirability of new plans can thus be dovetailed with computational accounts of plan generation. We developed both a theoretical framework for option evaluation in context, and an anytime algorithm that can be used by a computational agent for such evaluation.

Computing Plan Compatibility

Publications [1],[4],[21]

A second important plan-management task involves determining whether a new option is compatible with existing commitments. Another way to view this problem is to ask whether the new option can be merged into the existing set of plans. The problem of plan merging was previously studied by Q. Yang ("Intelligent Planning: A Decomposition and Abstraction Based Approach", Springer, New York, 1997), who developed a constraint satisfaction procedure to determine plan compatibility. This algorithm, COMBINE, takes as input two plans, identifies all the threats between them, constructs a constraint satisfaction problem (CSP) representing the threat resolution alternatives, and then solves the CSP to find some set of resolutions. Yang's approach, however, applied only to a limited class of plans, which were subject to severe limitations on expressive power.

We extended Yang's approach in two key ways. First, we generalized it so that it can handle plans with complex temporal constraints. Yang assumed "classical" plans, in which the only constraints on the timing of steps are relative ordering constraints. In contrast, we need to be able to reason about plans in which steps have specific assigned times (e.g., 2pm Wed.) and/or duration (e.g., 3 hours), and even temporal or duration intervals (e.g., Wed. between 2 and 5 pm, for a period of 1-2 hours). To handle such complex constraints, we developed efficient algorithms for solving various temporal reasoning problems, including Simple Temporal Problems (STPs) and Disjunctive Temporal Problems (DTPs). In fact, we showed experimentally that our algorithms for DTP solving achieved a two order-of-magnitude speed-up over the previous state-of-the-art algorithm.

Our second major extension involved making it possible to reason about the consistency of plans with conditional branches. Reasoning about the compatibility of conditional plans requires computing the possible intersections of future conditions, and checking for compatibility in each of these. We developed formalisms, Conditional Simple Temporal Problems (CSTPs) and Conditional Disjunctive Temporal Problems (CDTPs), and algorithms for solving these. Although solution in the general case is computationally intractable, we showed that there is a tractable subset of problems that corresponds to conditional plans.

Note that all our algorithms not only determine whether a new plan is consistent with a set of existing background plans, but also produce a set of any additional constraints that are necessary to guarantee consistency. Such "forced" constraints can then be adopted, along with the new plan, during plan update.

Contingency Selection

Publications [5],[7],[13],[20]

In dynamic and uncertain environments, it is impossible to form complex plans that are guaranteed to succeed: the space of contingent events is simply too large. Instead, an intelligent agent must make principled decisions about which possible contingencies to plan for in advance, and which to deal with, if necessary, at execution time. Most previous research on generating plans with conditional branches assumed small, static environments, and thus attempted to generate plans for all contingencies. In contrast, we developed an approach to contingency selection that makes use of the expected (dis)utility of failing to plan for a contingency.

Our algorithm has four notable features. First, we treat potential foreseeable execution failures as flaws, in the traditional POCL sense of a flaw. Specifically, we define a potential failure point to be any part of a plan that (a) involves a branching action, i.e., one whose outcome is uncertain, and (b) relies on a particular outcome of that action. Where classical planning algorithms consider open conditions and threats to be flaws, we add potential failure points into this set. Decisions about whether and when to handle each potential failure point can then be encoded as part of the search-control strategy.

Previous probabilistic/conditional planners have been severely limited by the fact that they do not know how to handle failure points to their advantage. For all but very small domains, the search space explodes quickly if plan failures are considered indiscriminately. We show how a principled selection of failure points can be performed within the framework of our algorithm.

Second, we include three logically distinct techniques to repair a potential failure point:

- *corrective repair*, originally introduced in the work on conditional planning, which involves reasoning about what to do if the desired outcome of a branching action does not occur;
- *preventive repair*, originally introduced in the work on probabilistic planning, which involves reasoning about how to help ensure that the desired outcome of a branching action will occur; and
- *replacement*, implemented by backtracking in the planning literature, which involves removing the branching action and replacing it with an alternative.

Third, as a result of the above techniques, the algorithm can generate plans that are conformant (covering each possible situation without utilizing an observation action), or conditional (with branches that are taken as a result of observation actions), or both. This feature allows the planner to perform a decision theoretic comparison of the plans that are generated using different strategies for handling possible failures. For instance, a conformant plan might be preferable when observation actions are costly, whereas a conditional plan might be preferable when the consequences of acting without looking are not favorable.

Fourth, our planner can generate conditional plans with merged branches: if two branches involve different steps at the beginning but the final steps are the same, the final part can be shared. Merging branches reduces the amount of planning required and thus improves efficiency.

Note that our work the generation of contingency plan fits with our previously described research on determining the consistency of plans with conditional branches.

Intelligent Execution Monitoring

Publications [3],[6],[8],[12],[15],[19]

As indicated above, in dynamic and uncertain environments, things will not always go "as planned." Thus, another key aspect of plan management is monitoring the execution of plans, so that replanning can be performed when needed. A central question is how to focus the sensing performed by such a system, so that it responds appropriately to relevant changes, but does not attempt to monitor all the changes that could possibly occur in the world. To achieve the required balance, we developed a new framework, called rationale-based monitors, which represent the features of the world state that are included in the plan rationale, i.e., the reasons for the planning decisions so far made. Rationale-based monitors capture information both about the plan currently under development and the alternative choices that were found but not pursued. We developed

plan transformations that result from the firing of a rationale-based monitor, for example when an alternative choice is detected. We implemented the rationale-based monitoring approach in two different planning systems: the Prodigy planner and a partial-order causal-link planner, and we explored the relative advantages of each planning approach for rationale-based monitoring. We have not yet extended the monitoring approach to the richly expressive plan structures developed and studied in the other components of this project, and thus the execution-monitoring mechanism in PMA is still rather unflexible. However, we believe that the theoretical foundations we developed are extensible. For example, there has been recent work that introduces uncertainty about action effects into a rationale-based monitoring-like framework (C. Boutilier, "Approximately Optimal Monitoring of Plan Preconditions," in Proceedings of the Conference on Uncertainty in AI, 2000.)

Overall Architecture and Prototype System

Publications [10], [11],[14],[16],[18]

In addition to our component studies, we also developed an integrated architecture for plan management, and a prototype plan management assistant (PMA). PMA is intended to be a "smart assistant", which helps a user manage a potentially large and complex set of plans in a dynamic setting. We developed a version of PMA for an academic user, whose routine procedures involve things like teaching courses, attending meetings, and overseeing the editing of papers. However, the central algorithms in the system are domain-independent and can be re-used for other types of users by modifying the knowledge base (indeed, in our NSF-funded project on cognitive orthotics for the elderly, we have done just this).

PMA is related to two major classes of software systems: personal electronic calendar systems and workflow systems. Commercially available electronic calendar systems, published by major software companies, essentially provide electronic interfaces to written calendars. They typically have advanced GUIs, and provide linkages to contact databases and email; some also provide capabilities for automated meeting scheduling. However, these systems suffer from a highly impoverished representation for activities: they can only model simple events and recurring simple events. Simple events are blocks of time with a single property---``free" or ``busy"; a ``free'' activity is allowed overlap with other ``free" activities, but a ``busy" activity cannot overlap with other activities. Recurring simple events are simple events that recur at regular intervals, e.g., every Tuesday from 4-5pm. Labels and textual information can be attached to each event, but these are not used in any sophisticated way by the system; they are stored only for the human user's information.

Workflow systems constitute another class of systems aimed at helping users manage their routine activities. In contrast to personal calendar systems, workflow systems employ richly structured representations of activities (or processes), and they use these representations to ensure that information and tasks flow to the appropriate people in an organization in a timely fashion. Modern workflow systems support document

management and coordination. On the other hand, they tend to have limited capabilities for handling uncertainty, for replanning when a problem is detected, and for reasoning about the relative value of alternative ways to perform a given task. PMA is being designed to include just these sorts of reasoning capabilities, using the techniques discussed above.

To illustrate the behavior of PMA, and show how it includes the types of reasoning processes already discussed, we describe a sample interaction with it. PMA has knowledge of the structured activities—the plans" or procedures—that its user typically performs. For instance, a PMA for use in a physician's office would know the steps involved in carrying out diagnostic procedures, preparing a patient for surgery, and handling insurance forms. The activity of preparing a patient for surgery might include, say, organizing a preliminary battery of tests, assembling and scheduling the surgical team, booking the operating room, etc. Many of these tasks would themselves decompose into structured activities: carrying out a single test might involve scheduling that test, tracking the lab work, entering the results into the patient's record, and calling the patient for follow-up work if necessary.

Imagine that a physician (or nurse) specifies the goal of performing a particular diagnostic test on a patient. PMA immediately posts commitments to various tasks pertaining to that goal in an internal knowledge base—the schedule. It also updates the graphical display that includes a calendar and to-do list. In this example, the posted commitments might include scheduling the test, obtaining the necessary background information before the test date, reminding the patient 48 hours before the test date, and so on. Once the user indicates that the test has been scheduled for a certain date—December 15, say—the temporal information associated with the related procedures will be updated accordingly; for example, a calendar entry will then appear reminding the user to notify the patient on December 13. Furthermore, if this test is just one of a battery of tests, and the scheduled December 15 date places it too near another test with which it might interfere, PMA will notice this conflict and notify the user, suggesting an alternative schedule that avoids the conflict. It may also suggest to the user that an operating room should be scheduled now, even though the actual deadline for the reservation has not yet occurred, because there is limited flexibility in the schedule to handle the situation should the operating rooms become unavailable at the desired time.

This scenario illustrates the main capabilities of PMA:

- The PMA user can commit to activities that have rich temporal and causal structure. She does not need to specify separate commitments to each component of the activity.

- The PMA user can make partial commitments: for instance, she can commit to performing a particular activity without yet specifying the exact time at which it will occur, or she can specify that she wants to commit to a particular goal, without yet specifying exactly which plan she will use to achieve that goal.

- When the user extends her commitments (e.g., by specifying a particular time or a particular plan for a goal), PMA propagates the new commitment to all affected parts of the activity. In the example above, when the user specifies that the test should be scheduled for Dec. 15, a patient reminder is automatically scheduled for Dec. 13.

- Whenever the user attempts to form a new commitment, PMA performs temporal and causal reasoning to determine whether it is consistent with the user's previous commitments. If PMA determines that certain additional constraints are required to ensure consistency, it notifies the user of those additional constraints, which we call "forced constraints." If PMA determines that there is a conflict between the new commitment and prior commitments, it suggests ways to resolve the conflict.

- PMA can assess the cost of executing a plan in the context of existing commitments, and notify the user if the cost fails to exceed some specified threshold.

- As time passes, PMA monitors the execution of the user's activities, and reminds the user when deadlines are approaching. It also reasons about the tightness of the schedule: for instance, if there is little slack at some future periods, PMA may suggest taking early action.

PMA has been implemented in Allegro Common Lisp for Windows. We implemented the first five capabilities just listed, using the representations and algorithms developed earlier in this project. We hope to introduce the remaining two capabilities to PMA in follow-on work. Our knowledge base so far contains plans for an academic user, rather than a medical one, but as noted above, the underlying mechanisms are domain-independent.

Publications supported by the Project

1. I. Tsamardinos, *Constraint-Based Temporal Reasoning Algorithms, with Applications to Planning*, University of Pittsburgh Ph.D. dissertation, 2001.

2. J. F. Horty and M. E. Pollack, "Evaluating New Options in the Context of Existing Plans," *Artificial Intelligence*, 127(2):199-220, 2001.

3. S. Ramakrishnan and M. E. Pollack, "Intelligent Monitoring in a Robotic Assistant for the Elderly," (Student Abstract), *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI)*, Aug. 2000.

4. I. Tsamardinos, M. E. Pollack, and J. F. Horty, "Merging Plans with Quantitative Temporal Constraints, Temporally Extended Actions, and Conditional Branches," Proceedings of the 5th International Conference on AI Planning Systems, Breckenridge, CO, April 2000. ***Winner of the Outstanding Student Paper Award.***

5. N. Onder, *Contingency Selection in Plan Generation*, University of Pittsburgh Ph.D. Dissertation, 1999.

6. C. McCarthy and M. E. Pollack, "Towards Focused Plan Monitoring: A Technique and an Application to Mobile Robots," *Autonomous Robots*, 9:71-81, 2000.

7. N. Onder and M. E. Pollack, "Generating Alternative Conditional Plans," AIPS Worskhop on Decision-Theoretic Planning, April 2000.

8. S. Ramakrishnan, "Simulation-Based Intelligent Reminding," University of Pittsburgh Intelligent Systems Program M.S. Project, 2000.

9. J. Hu, "Solving Multi-Agent Plan Refinement Problems using Local Search," University of Pittsburgh Intelligent Systems Program M.S. Project, 2000.

10. M. E. Pollack and J. F. Horty, "There's More to Life than Making Plans: Plan Management in Dynamic, Multi-Agent Environments," *AI Magazine* 20(4):71-84, 1999.

11. M. Georgeff, B. Pell, M. E. Pollack, M. Tambe, and M. Wooldridge, "The Belief-Desire-Intention Model of Agency," J. P. Muller, M. P. Singh, and A. S. Rao, eds. *Intelligent Agents V,* Springer Publishers, New York, 1999.

12. M. E. Pollack and C. McCarthy, "Towards Focused Plan Monitoring: A Technique and an Application to Mobile Robots," *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA),* November 1999.

13. N. Onder and M. E. Pollack, "Conditional, Probabilistic Planning: A Unifying Algorithm and Effective Search Control Mechanisms," 15th *National Conference on Artificial Intelligence (AAAI),* July 1999.

14. M. E. Pollack, I. Tsamardinos, and J. F. Horty, "Adjustable Autonomy for a Plan Management Agent," AAAI Spring Symposium on Adjustable Autonomy, March, 1999.

15. C. McCarthy, "Rationale-Based Monitoring: Application to Causal-Link Planning and Implementation in the Robotics Field," University of Pittsburgh Dept. of Computer Science M.S. Project, 1999.

16. J. F. Horty and M. E. Pollack, "Plan Management Issues for Cognitive Robotics: Project Overview," AAAI Fall Symposium on Cognitive Robotics, Orlando, FL, October 1998.

17. J. F. Horty and M. E. Pollack, "Evaluating Options in a Context," *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK),* Chicago, IL, July 1998.

18. M. E. Pollack, "Plan Generation, Plan Management, and the Design of Computational Agents (Abstract)" Proceedings *of the 3rd International Conference on Multi-Agent Systems*, Paris, France, July 1998.

19. M. M. Veloso, M. E. Pollack, and M. T. Cox, "Rationale-Based Monitoring for Planning in Dynamic Environments," Proceedings *of the 4th International Conference on AI Planning Systems,* Pittsburgh, PA, June, 1998.

20. N. Onder, M. E. Pollack, and J. F. Horty, "A Unified Algorithm for Conditional and Probabilistic Planning," AIPS Workshop on Integrating Planning, Scheduling, and Execution in Dynamic and Uncertain Environments, June 1998.

21. I. Tsamardinos, "Reformulating Temporal Plans for Efficient Execution," University of Pittsburgh Intelligent Systems Program M.S. Project, 1998.

Personnel supported by the Project

This project provided support for the Principal Investigator, Dr. Martha Pollack, as well as several graduate students in the Dept. of Computer Science and the Intelligent Systems Program at the University of Pittsburgh: Jun Hu, Atif Memon, Colleen McCarthy, Nilufer Onder, Sailesh Ramakrishnan, and Ioannis Tsamardinos. Not all of these students were supported throughout the entire duration of the contract: some graduated during the period of performance, and some had other sources of funding for certain periods (e.g., academic year fellowships). Project funding supported work on two dissertations (Onder and Tsamardinos) and on four M.S. projects (Hu, McCarthy, Ramakrishnan, and Tsamardinos).

Transitions and Follow-on Efforts

The software developed as part of this project is not yet ready for direct transition to a product. However, in a follow-on AFOSR contract (F49620-01-1-0066, "Increasing the Efficiency and Functionality of Plan Management Agents), we will be extending the foundational work done in the initial project along several dimensions, which should help close the gap between the research prototype and a system capable of being fielded. Of course, we also expect to produce foundational results on plan management and temporal reasoning in the new project. Additionally, we are now using many of the results we obtained in the initial project in a related project, being funded by the National Science Foundation, to develop technology that will assist elderly users with cognitive decline in managing their daily activities.